1. An apparatus for performing a fast pop operation from a random access cache memory, comprising:

a last-in-first-out (LIFO) memory, for storing a plurality of row values, said LIFO having a top entry for storing a newest row value; and

a multiplexer, comprising:

a first data input, coupled to receive said newest row value from said top entry;

a second data input, coupled to receive a row select portion of a memory address of an instruction accessing the cache memory;

an output, for providing a value for selecting a row of the cache memory; and

a selection input, for specifying a type of said instruction, wherein if said selection input specifies a pop instruction type, said multiplexer selects said first data input for provision on said output.

2. The apparatus of claim 1, wherein if said selection input specifies a load instruction type, said multiplexer selects said second data input for provision on said output.

3. The apparatus of claim 1, wherein if said selection input specifies an instruction type other than a pop instruction, said multiplexer selects said second data input for provision on said output.

4. The apparatus of claim 1, wherein each of said plurality of row values stored in said LIFO memory comprises a portion of a push instruction destination address.

5. The apparatus of claim 4, wherein said push instruction destination address is implied in said push instruction as an address relative to a stack pointer register value.

6. The apparatus of claim 4, wherein said newest row value comprises a portion of a newest push instruction destination address in said LIFO memory.

7. The apparatus of claim 1, wherein said memory address comprises a source address of a load instruction accessing the cache memory.

8. The apparatus of claim 7, wherein said load instruction source address is explicitly specified by said load instruction.

9. The apparatus of claim 1, further comprising:

an array of storage elements, coupled to receive said multiplexer output, having a plurality of rows each for storing one or more cache lines of data;

wherein said multiplexer output value specifies one of said plurality of rows.

10. The apparatus of claim 9, wherein one of said one or more cache lines of data in said one of said plurality of rows specified by said multiplexer output value contains destination data of a newest push instruction.

11. The apparatus of claim 10, wherein said newest push instruction comprises a most recently executed push instruction whose data has not yet been popped from a stack memory coupled to a microprocessor comprising the cache memory.

12. The apparatus of claim 9, wherein the cache memory outputs one of said one or more cache lines of data stored in one of said plurality of rows specified by said multiplexer output value, prior to determining whether a source address of said pop instruction hits in the cache memory.

13. The apparatus of claim 9, wherein said array of storage elements comprises one or more ways each for storing one of said one or more cache lines of data.

14. The apparatus of claim 1, further comprising:

a second LIFO memory, for storing a plurality of way values, said second LIFO having a top entry for storing a newest way value.

15. The apparatus of claim 14, further comprising:

a second multiplexer, comprising:

a first data input, coupled to receive said newest way value from said top entry of said second LIFO memory;

a second data input, coupled to receive a way select value;

an output, for providing a value for selecting a way of the cache memory; and

a selection input, for specifying said type of said instruction accessing the cache memory, wherein if said selection input specifies said pop instruction type, said second multiplexer selects said first data input for provision on said output.

16. The apparatus of claim 15, wherein if said second multiplexer selection input specifies a load instruction type, said second multiplexer selects said second data input for provision on said output.

17. The apparatus of claim 16, wherein said way select value comprises a way whose address tag matches a tag portion of a source address of said load instruction.

18. The apparatus of claim 15, wherein if said second multiplexer selection input specifies an instruction type other than said pop instruction, said second multiplexer selects said second data input for provision on said output.

19. The apparatus of claim 15, wherein each of said plurality of way values stored in said second LIFO memory specifies a way of the cache memory into which push instruction destination data was previously stored.

20. The apparatus of claim 15, wherein said row and said way selected by said first and second multiplexer outputs, respectively, specify a cache line stored in the cache memory containing source data for provision to said instruction.

21. The apparatus of claim 20, wherein if said selection input specifies a pop instruction type, said cache line specified by said row and said way is output by the cache memory, prior to determining whether a source address of said pop instruction hits in the cache memory.

22. The apparatus of claim 1, further comprising:

an offset, for specifying a location of data contained in a cache line of said row of the cache memory selected by said multiplexer output value.

23. The apparatus of claim 22, wherein said offset is incremented in response to a pop instruction.

24. The apparatus of claim 23, wherein if incrementing said offset causes said offset to overflow, the cache memory pops said newest row value off said LIFO memory.

25. The apparatus of claim 22, wherein said offset is decremented in response to a push instruction.

26. The apparatus of claim 1, further comprising:

comparison logic, coupled to receive said newest row value from said top entry, for comparing said newest row value with a portion of said memory address, said memory address comprising a source data address of said pop instruction.

27. The apparatus of claim 26, further comprising:

an exception output, coupled to said comparison logic, for indicating an exception condition if said newest row value does not match said portion of said memory address.

28. The apparatus of claim 1, wherein a computer program product comprising a computer usable medium having computer readable program code causes the apparatus, wherein said computer program product is for use with a computing device.

29. The apparatus of claim 1, wherein a computer data signal embodied in a transmission medium comprising computer-readable program code provides the apparatus.

30. A method for performing a fast pop operation from a random access cache memory, the method comprising:

storing data of a push instruction into a row of the cache specified by a row value;

pushing the row value onto a top entry of a LIFO memory in response to said storing;

receiving a request to read the cache memory after said pushing, wherein the request specifies a request type; and

reading the cache based on the row value stored in the top entry of the LIFO memory if the request type specifies a pop instruction type.

31. The method of claim 30, further comprising:

reading the cache based on a memory address specified by the request if the request type specifies a load instruction type.

32. The method of claim 30, wherein said storing further comprises storing the data of the push instruction into a way of the row of the cache, wherein the way is specified by a way value.

33. The method of claim 32, further comprising:

pushing the way value onto a top entry of a second LIFO memory in response to said storing.

34. The method of claim 33, wherein the first and second LIFO memories are a same LIFO memory.

35. The method of claim 33, further comprising:

reading the cache based on the way value stored in the top entry of the second LIFO memory if the request type specifies a pop instruction type.

36. The method of claim 30, further comprising:

determining whether the way value used in said reading correctly specified a way of the cache storing data specified by the pop instruction.

37. The method of claim 36, wherein said determining comprises:

comparing a portion of a source data address of the pop instruction with a tag value provided by the cache, wherein the tag value used in said comparing is selected based on the way value.

38. The method of claim 30, further comprising:

decrementing an offset value, in response to said storing, wherein the offset value specifies a location of the data within a cache line of the cache, wherein the cache line is within the row of the cache specified by the row value.

39. The method of claim 38, further comprising:

reading the cache based on the offset value if the request type specifies a pop instruction type.

40. The method of claim 30, further comprising:

determining whether the row value used in said reading correctly specified a row of the cache storing data specified by the pop instruction.

41. The method of claim 40, further comprising:

providing correct data to the pop instruction if the row value used in said reading incorrectly specified the row of the cache storing the data specified by the pop instruction.

42. The method of claim 40, further comprising:

generating an exception condition if the row value used in said reading incorrectly specified the row of the cache storing the data specified by the pop instruction.

43. The method of claim 40, wherein said determining comprises:

comparing a portion of a source data address of the pop instruction with the row value.

44. The method of claim 30, wherein said pushing is performed only if the push instruction data implicates a different cache line than implicated by an immediately previous push instruction.

45. The method of claim 30, further comprising:

popping the row value from the top entry of the LIFO memory if said reading comprises reading data from a cache line specified by the row value at an offset, and said offset specifies last data of the cache line.

46. A computer data signal embodied in a transmission medium, comprising:

computer-readable program code for providing an apparatus for performing a fast pop operation from a random access cache memory, said program code comprising:

first program code for providing a last-in-first-out (LIFO) memory, for storing a plurality of row values, said LIFO having a top entry for storing a newest row value; and

second program code for providing a multiplexer, comprising:

a first data input, coupled to receive said newest row value from said top entry; a second data input, coupled to receive a row select portion of a memory address;

a second data input, coupled to receive a row select portion of a memory address of an instruction accessing the cache memory;

an output, for providing a value for selecting a row of the cache memory; and

a selection input, for specifying a type of said instruction, wherein if said selection input specifies a pop instruction type, said multiplexer selects said first data input for provision on said output.